Name: Mohit Ranjit More

Class:TE-IT-T3

Subject: Operating System

Roll No: 3014

Pune Vidyarthi Griha's
College of Engineering and Technology & G. K. Pate
(Wani) Institute of Management
Approved by AICTE, DTE (Code: 6274) | Affiliated to SPPU, Pune | NAAC Second Cycle 'A' Grade

Assignment No: 7b

Problem Statement:

Inter-process Communication using Shared Memory using System V. Application to demonstrate: Client and Server Programs in which server process creates a shared memory segment and writes the message to the shared memory segment. Client process reads the message from the shared memory segment and displays it to the screen.

Solution:

Source Code
(Client)

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>

#define SHM_SIZE 1024  // Define the size of the shared memory

int main() {
   key_t key;
   int shmid;
   char *data;

   // Generate a unique key (same as server's key)
   key = ftok("shmfile", 65);
   if (key == -1) {
      perror("ftok");
      exit(1);
   }

   // Locate the shared memory segment created by the server
   shmid = shmget(key, SHM_SIZE, 0666);
   if (shmid == -1) {
      perror("shmget");
      exit(1);
   }

   // Attach the shared memory segment to the client's address space
   data = (char *)shmat(shmid, (void *)0, 0);
   if (data == (char *)(-1)) {
      perror("shmat");
      exit(1);
   }

   // Read the message from the shared memory
```

```
    printf("Message from shared memory: %s\n", data);

    // Detach from shared memory
    if (shmdt(data) == -1) {
        perror("shmdt");
        exit(1);
    }

    // Optionally, the client could remove the shared memory segment
    // shmctl(shmid, IPC_RMID, NULL);  // Uncomment to remove the shared memory

    return 0;
}
```

Source Code
(Server)

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>

#define SHM_SIZE 1024  // Define the size of the shared memory

int main() {
    key_t key;
    int shmid;
    char *data;

    // Generate a unique key
    key = ftok("shmfile", 65);
    if (key == -1) {
        perror("ftok");
        exit(1);
    }

    // Create the shared memory segment
    shmid = shmget(key, SHM_SIZE, 0666 | IPC_CREAT);
    if (shmid == -1) {
        perror("shmget");
        exit(1);
    }

    // Attach the shared memory segment to the server's address space
    data = (char *)shmat(shmid, (void *)0, 0);
    if (data == (char *)(-1)) {
        perror("shmat");
        exit(1);
    }

    // Write the message to the shared memory
```

```
    printf("Enter the message to write to shared memory: ");
    fgets(data, SHM_SIZE, stdin);

    // Detach from shared memory
    if (shmdt(data) == -1) {
        perror("shmdt");
        exit(1);
    }

    printf("Message written to shared memory.\n");

    return 0;
}
```

Source Output

```
mohit@mohit-VirtualBox:~$ cd mohit
mohit@mohit-VirtualBox:~/mohit$ gedit server.c
mohit@mohit-VirtualBox:~/mohit$ gedit client.c
mohit@mohit-VirtualBox:~/mohit$ gcc server.c -o server
mohit@mohit-VirtualBox:~/mohit$ gcc client.c -o client
mohit@mohit-VirtualBox:~/mohit$ ./server
ftok: No such file or directory
mohit@mohit-VirtualBox:~/mohit$ touch shmfile
mohit@mohit-VirtualBox:~/mohit$ ls -l shmfile
-rw-rw-r-- 1 mohit mohit 0 Oct 18 00:04 shmfile
mohit@mohit-VirtualBox:~/mohit$ ./server
Enter the message to write to shared memory: Hello , Mohit here from solapur
Message written to shared memory.
mohit@mohit-VirtualBox:~/mohit$ ./client
Message from shared memory: Hello , Mohit here from solapur

mohit@mohit-VirtualBox:~/mohit$
```